Qualitative Data Cleaning

Xu Chu
University of Waterloo
x4chu@uwaterloo.ca

Ihab F. Ilyas
University of Waterloo
ilyas@uwaterloo.ca

ABSTRACT

Data quality is one of the most important problems in data management, since dirty data often leads to inaccurate data analytics results and wrong business decisions.

Data cleaning exercise often consist of two phases: error detection and error repairing. Error detection techniques can either be quantitative or qualitative; and error repairing is performed by applying data transformation scripts or by involving human experts, and sometimes both.

In this tutorial, we discuss the main facets and directions in designing qualitative data cleaning techniques. We present a taxonomy of current qualitative error detection techniques, as well as a taxonomy of current data repairing techniques. We will also discuss proposals for tackling the challenges for cleaning "big data" in terms of scale and distribution.

1. INTRODUCTION

Enterprises have been acquiring large amounts of data from a variety of sources to build their own "Data Lakes", with the goal of enriching their data asset and enabling richer and more informed analytics. Data collection and acquisition often introduce errors in data, for example, missing values, typos, mixed formats, replicated entries of the same real-world entity, and violations of business rules. Not surprisingly, developing effective and efficient data cleaning solutions is a challenging venue and is rich with deep theoretical and engineering problems.

Data cleaning usually consists of two phases: error detection and error repairing. Error detection techniques can be either quantitative or qualitative. Quantitative error detection techniques often involve statistical methods to identify abnormal behaviors and errors [22] (e.g., *"a salary that is three standard deviation away from the mean salary is an error"*). Quantitative error detection has been mostly studied in the context of outlier detection [1]. On the other hand, qualitative error detection techniques rely on descriptive approaches to specify patterns or constraints of a legal data instance, and hence identify those data that violate those patterns or constraints as errors. For example, a descriptive statement about an employee database is *"there cannot exist two employees of the same level, the one who is located in NYC is earning less than the one not in NYC"*; if we find two such employees, then we are certain that there is an error in at least one of the employees' records.

Error repairing is performed by either applying data transformation scripts, which are usually generated according to the process used for error detection, or by involving human experts in a principled manner, or by a combination of both.

Quantitative data cleaning techniques have been heavily studied in multiple surveys [1, 30, 22] and tutorials [27, 9], but less so for qualitative data cleaning techniques. Given the recent surge of papers on pattern-based or constraints-based data cleaning systems [7, 13, 19, 16, 32, 12, 37, 14, 3, 17, 35], we believe that a tutorial that sheds light on these proposals and how they relate to each other is due. We focus in this tutorial on qualitative data cleaning techniques, and we propose taxonomies to classify different error detection and error repairing techniques. For every type of error detection and error repairing techniques, we will use one or more examples to illustrate. We will also introduce challenges raised by "big data" settings, and explore the current available data cleaning proposals that aim at tackling those challenges.

The audience of this tutorial includes researchers and practitioners who are interested in data quality and data cleaning. Besides a basic understanding of database technology, there is no prerequisites. The tutorial is 1.5 hours. Most of the materials of this tutorial comes from a survey we published in *Foundations and Trends in Databases* [23].

2. TUTORIAL OVERVIEW

We will start the tutorial by giving an overview of the area of data quality management. We will then give a motivating example highlighting many different data quality problems, such as duplicates, missing values, integrity constraints violations, and outliers, and thus stressing the importance of effective data cleaning. We will then scope this tutorial to focus on qualitative data cleaning techniques.

Since data cleaning usually consists of two stages: error detection and error repairing, we will discuss a variety of techniques for qualitative error detection (Section 2.1), as well as various techniques for error repairing (Section 2.2). We will then discuss the challenges raised by cleaning big data (Section 2.3). Finally, we will summarize the trends in data cleaning, and provide a list of interesting future work in the area (Section 2.4).

Proceedings of the VLDB Endowment, Vol. 9, No. 13
Copyright 2016 VLDB Endowment 2150-8097/16/09.

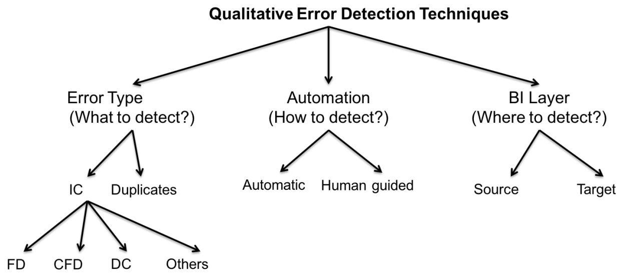

Figure 1: Classification of qualitative error detection techniques.

2.1 Qualitative Error Detection

Given a dirty database instance, the first step toward cleaning the database is to detect and surface anomalies or errors. Figure 1 depicts the classification we adopt to categorize the current qulitative error detection techniques. In the following, we discuss our classification. The three adopted dimensions capture the three main questions involved in detecting errors in a database.

• *Error Type (What to Detect?)* Qualitative error detection techniques can be classified according to which type of errors are captured. In other words, what languages are used to describe patterns or constraints of a legal data instance. A large body of work uses integrity constraints (ICs), a fractional of first order logic, to capture data quality rules that the database should conform to, including functional dependencies (FDs) [6], and denial constrains (DCs) [13]. While duplicate records can be considered a violation of an integrity constraint (key constraint), we recognize the large body of work that focuses on this problem and we discuss it as a separate error type from other types of integrity constraints.

Manual designing such ICs or patterns require great domain expertise, and is time-consuming, automatic discovery techniques are essential, and have been proposed for various ICs [12]. We classify the IC discovery techniques into schema-driven approaches and instance driven approaches, and we will discuss and compare these two approaches.

• *Automation (How to Detect?)* We classify proposed approaches according to whether and how humans are involved in the anomaly detection process. Most techniques are fully automatic, for example, detecting violations of functional dependencies [6], while other techniques involve humans, for example, to identify duplicate records [33].

• *Business Intelligence Layer (Where to Detect?)* Errors can happen in all stages of a business intelligence (BI) stack, for example, errors in the source database are often propagated through the data processing pipeline. While most anomaly detection techniques detect errors in the original database, some errors can only be discovered much later in the data processing pipeline [8], where more semantics and business logics becomes available, for example, constraints on total budget can only be enforced after aggregating cost and expenses.

Table 1 shows a sample of anomaly detection techniques, which cover all categories of the proposed taxonomy. We will give one or more example error detection techniques in each category in detail in the tutorial.

	Error Type What		Automation How		BI Layer Where	
	IC	Data deduplication	Automatic	Human involved	Source	Target
FDs value modification [6]	✓		✓		✓	
Holistic data cleaning [13]	✓		✓		✓	
CrowdER [33]		✓		✓	✓	
Corleone [20]		✓		✓	✓	
Causality Analysis [28]	✓			✓		✓
Scorpion [36]	✓			✓		✓
DBRx [8]	✓		✓			✓

Table 1: A sample of qualitative error detection techniques.

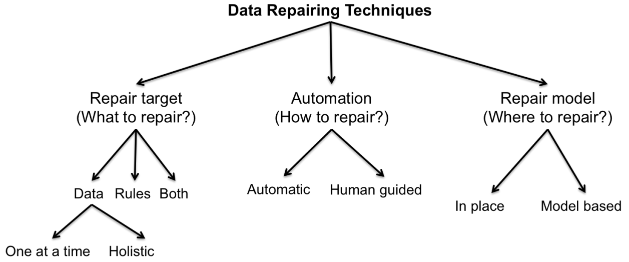

Figure 2: Classification of data repairing techniques.

2.2 Error Repairing

Given a relational database instance I of schema R and a set of data quality requirements expressed in a variety of ways, data repairing refers to the process of finding another database instance I' that conforms to the set of data quality requirements. A plethora of data repairing techniques have been proposed. Figure 2 depicts the classification we adopt to categorize the proposed data repairing techniques. In the following, we discuss our classification dimensions, and their impact on the design of underlying data repairing techniques. The three adopted dimensions capture the three main questions involved in repairing an erroneous databases:

• *Repair Target (What to Repair?)* Repairing algorithms make different assumptions about the data and the quality rules: (1) trusting the declared integrity constraints, and hence, only data can be updated to remove errors [13]; (2) trusting the data completely and allowing the relaxation of the constraints [21], for example, to address schema evolution and obsolete business rules; and finally (3) exploring the possibility of changing both the data and the constraints [4]. For techniques that trust the rules, and change only the data, they can be further divided according to the driver to the repairing exercise, that is, what types of errors they are targeting. A majority of techniques repair the data with respect to one type of errors only (one at a time), while other emerging techniques consider the interactions among multiple types of errors and provide a holistic repair of the data (holistic).

• *Automation (How to Repair?)* We classify proposed approaches with respect to the tools used in the repairing process. More specifically, we classify current repairing approaches according to whether and how humans are involved. Some techniques are fully automatic, for example, by modifying the database, such that the distance between the original database I and the modified database I' is minimized according to some cost function. Other techniques

	Repair target What				Automation How		Repair model Where	
	Data - One at a time	Data - Holistic	Rules	Both	Automatic	Human involved	In place	Model based
FDs value modification [6]	✓				✓		✓	
FDs hypergraph [25]	✓				✓		✓	
CFDs value modification [15]	✓				✓		✓	
Holistic data cleaning [13]		✓			✓		✓	
LLUNATIC [19]		✓			✓		✓	
Record matching and data repairing [18]		✓			✓		✓	
NADEEF [16]		✓			✓		✓	
Generate optimal tablaux [21]			✓		✓		✓	
Unified repair [10]				✓	✓		✓	
Relative trust [4]				✓	✓		✓	
Continuous data cleaning [32]				✓	✓		✓	
Potter's Wheel [29]	✓					✓	✓	
GDR [37]	✓					✓	✓	
KATARA [14]	✓					✓	✓	
DataTamer [31]	✓					✓	✓	
Editing rules [17]	✓					✓	✓	
Sampling FDs repairs [3]	✓				✓			✓
Sampling Duplicates [5]	✓				✓			✓

Table 2: A sample of data repairing techniques.

involve humans in the repairing process either to verify the fixes, to suggest fixes, or to train machine learning models to carry out automatic repairing decisions [37].

• *Repair Model (Where to Repair?)* We classify proposed approaches based on whether they change the database in-situ, or build a model to describe the possible repairs. Most proposed techniques repair the database in place, thus destructing the original database. For none in-situ repairs, a model is often built to describe the different ways to repair the underlying database. Queries are answered against these repairing models using, for example, sampling from all possible repairs and other probabilistic query answering mechanisms [3].

Table 2 shows a sample of data repairing techniques using the taxonomy. We will discuss one or more example error repairing techniques in each category in detail in the tutorial.

2.3 Big Data Cleaning Challenges

With the advent of big data era, data cleaning has come more important and challenging than ever. Due to the sheer volume of generated data, and the fast velocity of arriving data, data cleaning activities need to be performed in a scalable and timely manner, and at the same time cope with the increasing variety of data sources. In this section, we discuss various algorithmic and systematic approaches in cleaning big data, including blocking for duplicate detection [2], sampling for data cleaning [34], and distributed data cleaning [26, 24, 11].

2.4 Trends and Future Work

Data quality and data cleaning are becoming more important than ever, with direct and timely needs in the Big Data era. Data cleaning is the first line of defense in extracting value from the huge amounts of heterogeneous, incomplete, and continuously growing data sets. We envision multiple future work directions, we list some of them in the following:

• *Error Detection.* While we have discussed several ways to detect anomalies in the data, many data errors may still remain undetected. One direction is to devise more expressive integrity constraint languages that allow data owners to easily specify data quality rules and to effectively involve human experts in anomaly detection.

• *Master data curation.* To perform reliable data repairing, master data often needs to be referenced. However, existing master data sources, such as knowledge bases, often cannot provide a comprehensive coverage for the data to be repaired. Automatic creation and maintenance of relevant master and authoritative data catalogs are essential tasks in enabling high-quality repairs.

• *Human involved data repairing.* Although much research has been done about involving humans to perform data deduplication, involving humans in other data cleaning tasks, such as repairing IC violations is yet to be explored.

• *Scalability.* Large volumes of data render most current techniques unusable in real settings. The obvious trade-off between accuracy and performance has to be taken more seriously in designing the next generation cleaning algorithms that take time and space budget into account. Example tools include sampling, and approximate cleaning algorithms, with clear approximation semantics that can be leveraged by analytics applications.

• *Semi-structured and unstructured data.* A significant portion of data is residing in semi-structured formats, e.g., JSON, and unstructured formats, e.g., text documents. Data quality problems for semi-structured and unstructured data remain unexplored.

3. CONCLUSION

In this tutorial, we shed some light on some of the foundational aspects and trends in qualitative data cleaning efforts. We primarily focus on the two phases of data cleaning: error detection and repairing. For error detection, we provide a classification for qualitative error detection techniques based on What, How and Where to detect the errors; for data repairing, we also provide a classification for data repairing techniques based on What, How and Where to repair the data. There are multiple important directions to pursue as highlighted.

4. BIOGRAPHIES

Xu Chu

Xu Chu is a PhD student in the Cheriton School of Computer Science at University of Waterloo. His main research interests are data quality and data cleaning. He won the prestigious Microsoft Research PhD fellowship in 2015. Xu has also received Cheriton Fellowship from the University of Waterloo 2013-2015.

Ihab F. Ilyas

Ihab Ilyas is a professor in the Cheriton School of Computer Science at the University of Waterloo. He received his PhD in computer science from Purdue University, West Lafayette. His main research is in the area of database systems, with special interest in data quality, managing uncertain data, rank-aware query processing, and information extraction. Ihab is a recipient of the Ontario Early Researcher Award (2009), a Cheriton Faculty Fellowship (2013), an NSERC Discovery Accelerator Award (2014), and a Google Faculty Award (2014), and he is an ACM Distinguished Scientist. Ihab is a co-founder of Tamr, a startup focusing on large-scale data integration and cleaning. He serves on the VLDB Board of Trustees, and he is an associate editor of the ACM Transactions of Database Systems (TODS).

5. REFERENCES

[1] C. C. Aggarwal. *Outlier Analysis*. Springer, 2013.

[2] R. Ananthakrishna, S. Chaudhuri, and V. Ganti. Eliminating fuzzy duplicates in data warehouses. In *PVLDB*, pages 586–597, 2002.

[3] G. Beskales, I. F. Ilyas, and L. Golab. Sampling the repairs of functional dependency violations under hard constraints. *PVLDB*, 3(1-2):197–207, 2010.

[4] G. Beskales, I. F. Ilyas, L. Golab, and A. Galiullin. On the relative trust between inconsistent data and inaccurate constraints. In *ICDE*, pages 541–552, 2013.

[5] G. Beskales, M. A. Soliman, I. F. Ilyas, and S. Ben-David. Modeling and querying possible repairs in duplicate detection. *PVLDB*, pages 598–609, 2009.

[6] P. Bohannon, W. Fan, M. Flaster, and R. Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *SIGMOD*, pages 143–154. ACM, 2005.

[7] P. Bohannon, W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for data cleaning. In *ICDE*, pages 746–755, 2007.

[8] A. Chalamalla, I. F. Ilyas, M. Ouzzani, and P. Papotti. Descriptive and prescriptive data cleaning. In *SIGMOD*, pages 445–456, 2014.

[9] S. Chawla and P. Sun. Outlier detection: Principles, techniques and applications. In *PAKDD*, 2006.

[10] F. Chiang and R. J. Miller. A unified model for data and constraint repair. In *ICDE*, pages 446–457, 2011.

[11] X. Chu, I. F. Ilyas, and P. Koutris. Distributed Data Deduplication. *PVLDB*, 9(11), 2016.

[12] X. Chu, I. F. Ilyas, and P. Papotti. Discovering denial constraints. *PVLDB*, 6(13):1498–1509, 2013.

[13] X. Chu, I. F. Ilyas, and P. Papotti. Holistic data cleaning: Putting violations into context. In *ICDE*, pages 458–469, 2013.

[14] X. Chu, J. Morcos, I. F. Ilyas, M. Ouzzani, P. Papotti, N. Tang, and Y. Ye. KATARA: A data cleaning system powered by knowledge bases and crowdsourcing. In *SIGMOD*, pages 1247–1261, 2015.

[15] G. Cong, W. Fan, F. Geerts, X. Jia, and S. Ma. Improving data quality: Consistency and accuracy. In *PVLDB*, pages 315–326. VLDB Endowment, 2007.

[16] M. Dallachiesa, A. Ebaid, A. Eldawy, A. Elmagarmid, I. F. Ilyas, M. Ouzzani, and N. Tang. Nadeef: a commodity data cleaning system. In *SIGMOD*, pages 541–552, 2013.

[17] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Towards certain fixes with editing rules and master data. *PVLDB*, 3(1-2):173–184, 2010.

[18] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Interaction between record matching and data repairing. In *SIGMOD*, pages 469–480. ACM, 2011.

[19] F. Geerts, G. Mecca, P. Papotti, and D. Santoro. The llunatic data-cleaning framework. *PVLDB*, 6(9):625–636, 2013.

[20] C. Gokhale, S. Das, A. Doan, J. F. Naughton, N. Rampalli, J. Shavlik, and X. Zhu. Corleone: Hands-off crowdsourcing for entity matching. In *SIGMOD*, pages 601–612, 2014.

[21] L. Golab, H. Karloff, F. Korn, D. Srivastava, and B. Yu. On generating near-optimal tableaux for conditional functional dependencies. *PVLDB*, 1(1):376–390, 2008.

[22] J. M. Hellerstein. Quantitative data cleaning for large databases. *United Nations Economic Commission for Europe (UNECE)*, 2008.

[23] I. F. Ilyas and X. Chu. Trends in cleaning relational data: Consistency and deduplication. *Foundations and Trends in Databases*, 5(4):281–393, 2015.

[24] Z. Khayyat, I. F. Ilyas, A. Jindal, S. Madden, M. Ouzzani, P. Papotti, J.-A. Quiané-Ruiz, N. Tang, and S. Yin. Bigdansing: A system for big data cleansing. In *SIGMOD*, pages 1215–1230, 2015.

[25] S. Kolahi and L. V. S. Lakshmanan. On approximating optimum repairs for functional dependency violations. In *ICDT*, pages 53–62, 2009.

[26] L. Kolb, A. Thor, and E. Rahm. Dedoop: efficient deduplication with hadoop. *PVLDB*, 5(12):1878–1881, 2012.

[27] H.-P. Kriegel, P. Kröger, and A. Zimek. Outlier detection techniques. In *Tutorial at SIGKDD*, 2010.

[28] A. Meliou, W. Gatterbauer, S. Nath, and D. Suciu. Tracing data errors with view-conditioned causality. In *SIGMOD*, pages 505–516, 2011.

[29] V. Raman and J. M. Hellerstein. Potter's wheel: An interactive data cleaning system. In *VLDB*, pages 381–390, 2001.

[30] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*, volume 589. John Wiley & Sons, 2005.

[31] M. Stonebraker, D. Bruckner, I. F. Ilyas, G. Beskales, M. Cherniack, S. B. Zdonik, A. Pagan, and S. Xu. Data curation at scale: The data tamer system. In *CIDR*, 2013.

[32] M. Volkovs, F. Chiang, J. Szlichta, and R. J. Miller. Continuous data cleaning. In *ICDE*, pages 244–255, 2014.

[33] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *PVLDB*, 5(11):1483–1494, 2012.

[34] J. Wang, S. Krishnan, M. J. Franklin, K. Goldberg, T. Kraska, and T. Milo. A sample-and-clean framework for fast and accurate query processing on dirty data. In *SIGMOD*, pages 469–480, 2014.

[35] J. Wang and N. Tang. Towards dependable data repairing with fixing rules. In *SIGMOD*, pages 457–468. ACM, 2014.

[36] E. Wu and S. Madden. Scorpion: Explaining away outliers in aggregate queries. *PVLDB*, 6(8):553–564, 2013.

[37] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas. Guided data repair. *PVLDB*, 4(5):279–289, 2011.